

# Manual de Instalación

Este manual te guiará en los pasos necesarios para instalar y configurar el proyecto desde cero. Asegúrate de seguir cada sección con atención para evitar problemas durante la instalación.

## Requisitos Previos

Antes de comenzar, asegúrate de tener instalados los siguientes componentes en tu sistema:

- **Node.js** (versión 18 o superior): Puedes descargarlo desde [Node.js](https://nodejs.org/).
- **npm** (incluido con Node.js).
- **SQL Server**: Instalado localmente o mediante un servidor accesible.
- **Git**: Para clonar el repositorio. Descárgalo desde [Git](https://git-scm.com/).

Nota: Este proyecto fue desarrollado en un entorno con las versiones mencionadas anteriormente. Aunque otras versiones pueden funcionar, es recomendable usar estas para evitar posibles problemas de compatibilidad.

## Paso 1: Clonar el Repositorio

Clona el repositorio en tu máquina local. Abre una terminal y ejecuta el siguiente comando:

```
git clone https://github.com/christian-lopezxd/TaskifyAPI.git
```

Una vez clonado, entra en el directorio del proyecto:

```
cd TaskifyAPI/taskify_api
```

## Paso 2: Configurar Variables de Entorno

Este proyecto utiliza un archivo `.env` para gestionar las configuraciones sensibles, como las credenciales de la base de datos. En el directorio principal del proyecto, crea un archivo `.env` y copia las variables necesarias:

```
DATABASE_URL="sqlserver://<HOST>:<PORT>;database=taskifybd;user=<USER>;password=<PASSWORD>;encrypt=true;trustServerCertificate=true"
```

Reemplaza los valores `<USER>`, `<PASSWORD>`, `<HOST>`, `<PORT>` con tus credenciales y la información de tu servidor SQL Server.

**Importante:** Asegúrate de que el servidor SQL está configurado para aceptar conexiones externas y que tienes acceso a las credenciales proporcionadas.

## Paso 3: Instalar Dependencias

Con la terminal en el directorio raíz del proyecto, instala las dependencias necesarias ejecutando el siguiente comando:

```
npm install
```

Este comando instalará todos los paquetes requeridos que se encuentran en el archivo `package.json`.

## Paso 4: Configurar Prisma (ORM)

Este proyecto utiliza Prisma entonces necesitas generar el cliente de Prisma y migrar el esquema. Sigue estos pasos:

### 1. Generar el Cliente de Prisma:

```
npx prisma generate
```

### 2. Ejecutar Migraciones

```
npx prisma migrate dev --name inicial
```

Esto creará las tablas y estructuras definidas en el esquema de Prisma en la base de datos configurada.

## Paso 5: Ejecutar el Proyecto en Modo de Desarrollo

Para iniciar el servidor en modo de desarrollo, usa el siguiente comando:

```
npm run start:dev
```

## Paso 6: Verificar la Instalación

Para asegurarte de que la instalación fue exitosa:

1. **Verifica la conexión a la API** abriendo un navegador para acceder a <http://localhost:3000/api-docs>.
  2. **Prueba de Rutas:** Accede a las rutas base configuradas en el proyecto (por ejemplo, [/users](#) y [/tasks](#)) para verificar que la aplicación responde correctamente.
  3. **Revisar la consola:** Asegúrate de que no aparezcan errores en la consola mientras el proyecto está corriendo.
- 

## Resolución de Problemas Comunes

### Error: "Cannot find module..."

Si encuentras este error al iniciar el proyecto, asegúrate de haber ejecutado `npm install` para instalar todas las dependencias.

### Problema de Conexión con la Base de Datos

- Verifica que las credenciales y la URL en el archivo `.env` sean correctas.
- Asegúrate de que el servidor de base de datos esté activo y accesible desde tu entorno de desarrollo.

## Error en Prisma: "Database does not exist..."

Si Prisma arroja un error indicando que la base de datos no existe, asegúrate de que el nombre de la base de datos sea correcto y que la migración `npx prisma migrate dev` se haya ejecutado correctamente.